

# GNOME

Base GNOME packages for the full GNOME experience. Bundle with other packages to prevent package conflicts providing the same functionality.

**TIP:** Include any and all packages you want installed in a list to `pacman`. That way `pacman` will resolve package dependencies correctly and not install packages that would cause conflicts with other packages later on in the setup; e.g. the `gnome` group installs `pulseaudio`, but `pulseaudio` and `pipewire` (see below) are conflicting packages, meaning they can't both be installed at the same time prompting you to remove one or the other. Explicitly selected packages take precedence over packages auto-selected via dependencies.

```
pacman -S gnome gnome-extra
```

## Setting up display manager

### Start GDM on boot

Start the GNOME Display Manager (GDM) on boot to be presented with a graphical login screen.

```
systemctl enable gdm
```

### When using NVIDIA proprietary drivers

For the longest time NVIDIA only supported their EGLStreams interface for Wayland sessions. Despite GNOME having support for both EGLStreams and the more popular GBM interface, the GNOME Display Manager disables the Wayland session via a `udev` rule, if it detects the proprietary driver is in use, to prevent problems with the login screen not showing.

To force enable GNOME's Wayland session even with the proprietary NVIDIA driver installed, check the following files:

- `/etc/gdm/custom.conf`: Make sure the line `WaylandEnable=false` is **commented out** (should be by default)
- `/usr/lib/udev/rules.d/61-gdm.rules`: Rename the file and create a symbolic link to `/dev/null`

```
In -s /dev/null /usr/lib/udev/rules.d/61-gdm.rules
```

Keep in mind that Wayland depends on Kernel Mode Setting to function properly, so it is necessary to include the appropriate kernel modules in the kernel image and setting the kernel commandline parameter to enable KMS support for the proprietary NVIDIA driver!

See [Graphics Cards](#) on how to set up early KMS with the proprietary NVIDIA driver.

## Set Keymap for GDM

**NOTE:** Executing this command while `chroot`ed into an installation will produce an error that the locale could not be found. Set after rebooting the system, press `CTRL + ALT + F3` when GDM shows up (or any F-key between 2 and 7) to switch tty, log in via the command line and execute the command as `root`.

```
localectl set-x11-keymap de
```

See instructions at [Plymouth page](#) on how to set up Plymouth.

## Generate well-known user directories

```
xdg-user-dirs-update
```

## Misc additional packages

Additional packages you might want:

Name	Description
<code>gthumb</code>	Image viewer with simple editing capabilities
<code>lollypop</code>	Music player for GNOME
<code>seahorse</code>	Secrets manager (login credentials, SSH keys, GPG keys)

Name	Description
fwupd	Firmware update manager; allows UEFI capsule updates in GNOME Software if supported by firmware
gnome-software-packagekit-plugin	Manage Arch packages in GNOME Software

```
pacman -S gthumb lollypop seahorse fwupd gnome-software-packagekit-plugin
```

# GNOME Keyring

Gnome Keyring is a useful tool for securely storing and managing passwords, SSH keys, and other sensitive information.

As `gnome-keyring` is already a member of the `gnome` package group, it should already be installed.

To manage the contents of `gnome-keyring` install `seahorse`:

```
pacman -S seahorse
```

## SSH Keys

You can use `gnome-keyring` to store the passphrases of your SSH keys for passwordless authentication. However, the systemd user service file does not include the `ssh` component.

To include the `ssh` component, edit the systemd unit file:

```
systemctl edit --user gnome-keyring-daemon
```

This opens an editor with a temporary drop-in file. In here you can override the parameters of the systemd unit file:

**NOTE:** To properly replace a parameter, it first needs to be empty and then repeated on the next line with the value you want it to have. Additionally, setting the `SSH_AUTH_SOCK` environment variable will make applications aware of an already unlocked SSH key.

```
[Service]
ExecStart=
ExecStart=/usr/bin/gnome-keyring-daemon --foreground --components="pkcs11,secrets,ssh" --control-
directory=%t/keyring
```

Save the file and close the editor. It will get automatically reloaded by systemd if necessary.

In order to reveal to applications that an SSH key has already been unlocked, set an environment variable for the current user:

```
echo SSH_AUTH_SOCKET=$XDG_RUNTIME_DIR/keyring/ssh >> ~/.config/environment.d/envvars.conf
```

Re-login for the changes to take effect.

# Remove potentially unwanted packages

## GNOME Dev Tools

```
pacman -Rsc accerciser devhelp glade gnome-builder sysprof
```

## User Software

```
pacman -Rsc gnome-recipes
```

## Games

```
pacman -Rsc five-or-more four-in-a-row gnome-chess gnome-klotski gnome-mahjongg gnome-mines gnome-nibbles gnome-robots gnome-robots gnome-sudoku gnome-taquin gnome-tetravex hitori iagno lightsoff polari quadrapassel swell-foop tali
```

# Customize GDM (wallpaper, logo, message)

1. Create directories:

```
mkdir -p /etc/dconf/profile
mkdir -p /etc/dconf/db/gdm.d/
```

## 2. Create config files

```
touch /etc/dconf/profile/gdm
touch /etc/dconf/db/gdm.d/01-login-screen
```

## 3. Contents of `/etc/dconf/profile/gdm`

```
user-db:user
system-db:gdm
file-db:/usr/share/gdm/greeter-dconf-defaults
```

## 4. Contents of `/etc/dconf/db/gdm.d/01-login-screen`

```
[org/gnome/login-screen]
banner-message-enable=true
banner-message-text='Lorem ipsum dolor sit amet'
logo='/path/to/image.file'
[org/gnome/desktop/background]
picture-uri='file:///path/to/background.jpg'
```

## 5. Update gconf to apply configs

```
dconf update
```

---

Revision #21

Created 31 August 2021 12:32:03 by Sebin

Updated 26 March 2023 04:28:27 by Sebin